Inexact Barzilai-Borwein Method for Saddle Point Problems *

Yi-Qing Hu[†] and Yu-Hong Dai[†‡]

Abstract

This paper considers the inexact Barzilai-Borwein algorithm applied to saddle point problems. To this aim, we study the convergence properties of the inexact Barzilai-Borwein algorithm for symmetric positive definite linear systems. Suppose that g_k and $\tilde{g}_k$ are the exact residual and its approximation of the linear system at the k-th iteration, respectively. We prove the R-linear convergence of the algorithm if $\|\tilde{g}_k - g_k\| \leq \eta \|\tilde{g}_k\|$ for some small $\eta > 0$ and all k. To adapt the algorithm for solving saddle point problems, we also extend the R-linear convergence result to the case when the right hand term $\|\tilde{g}_k\|$ is replaced by $\|\tilde{g}_{k-1}\|$. Although our theoretical analyses cannot provide a good estimate to the parameter η, in practice we find that η can be as large as the one in the inexact Uzawa algorithm. Further numerical experiments show that the inexact Barzilai-Borwein algorithm performs well for the tested saddle point problems.

Keywords. Saddle point problem, Uzawa algorithm, Barzilai-Borwein method, R-linear convergence.

---

*This work was supported by the Chinese NSF grants 10571171, 40233029 and 10171104.

[†]State Key Laboratory of Scientific and Engineering Computing, Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, P.O. Box 2719, Beijing 100080, P.R. China. E-mail Addresses: {*huyq, dyh*}@*lsec.cc.ac.cn*

[‡]Corresponding author.

1. Introduction

We consider the saddle point problem:

$$\begin{pmatrix} A & B^T \\ B & -C \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f \\ h \end{pmatrix} , \tag{1.1}$$

where $A \in \Re^{n\times n}$ is symmetric positive definite and $C \in \Re^{m\times m}$ is symmetric positive semidefinite. This kind of problem arises frequently from the discretization of elasticity problems, Stokes equations, and sometimes linearizations of Navier-Stokes equations. It also has a close relation to nonlinear programming since the problem of minimizing a convex quadratic subject to linear constraints can be converted into the form (1.1). There have been many methods developed for problem (1.1), see recent survey paper [4] and book [17]). In this paper, we are interesting in a classic algorithm that is due to Uzawa [1]. It can be written as

Algorithm 1.1 *(Uzawa)*

Step 1. Initialize $k = 0$ *and pick some* $p_0 \in \Re^m$*;*

Step 2. Solve $Au_{k+1} = f - B^T p_k$ *for* $u_{k+1} \in \Re^n$*;*

Step 3. Calculate $p_{k+1} = p_k - \alpha(Cp_k - Bu_{k+1} + h)$*;*

Step 4. If not convergent, set $k = k + 1$ *and go to Step 1.*

The elimination of u_{k+1} in the calculation of p_{k+1} leads to the iteration

$$p_{k+1} = p_k - \alpha[(BA^{-1}B^T + C)p_k - (BA^{-1}f - h)] . \tag{1.2}$$

Therefore the Uzawa algorithm is a fixed-parameter first-order Richardson iterative method [27] applied to the linear system

$$(BA^{-1}B^T + C)p = BA^{-1}f - h . \tag{1.3}$$

In the context of optimization, the algorithm can be regarded as a fixed stepsize gradient method for the problem of minimizing a convex quadratic

$$\min \quad \frac{1}{2} p^T \bar{A} p - \bar{b}^T p , \tag{1.4}$$

where $\bar{A} = BA^{-1}B^T + C$ and $\bar{b} = (BA^{-1}f - h)$.

The choice of the parameter α is important to the efficacy of the Uzawa algorithm. Elman and Golub [16] proposed the following choice

$$\alpha = \frac{2}{\lambda_1 + \lambda_n}, \tag{1.5}$$

where λ_1 and λ_n are the minimal and maximal eigenvalues of the matrix $\bar{A}$, respectively. This choice is optimal in the sense that it minimizes the spectral radius of the matrix $I - \alpha\bar{A}$. Since the eigenvalues λ_1 and λ_n are not known to the users in general, Dai and Yang [14] chose the stepsize as follows

$$\alpha_k = \frac{\|g_k\|}{\|\bar{A}g_k\|} \ , \tag{1.6}$$

where $g_k = \bar{A}p_k - \bar{b}$ and $\|\cdot\|$ is the two-norm. They proved that this sequence of $\{\alpha_k\}$ tends to the value in (1.5). In practical computations, however, the gradient method with either (1.5) or (1.6) resembles the steepest descent method (see Cauchy [10]), where

$$\alpha_k^{SD} = \frac{g_k^T g_k}{g_k^T \bar{A} g_k} \ . \tag{1.7}$$

They all become very slow as the condition number of the matrix $\bar{A}$ deteriorates. Consequently, the use of the Uzawa algorithm is usually with some preconditioning technique. The Uzawa algorithm has received much attention from the numerical linear algebra community, for example see [8], [9] and [22].

In 1988, Barzilai and Borwein [3] proposed a different choice for the stepsize in the gradient method. Their basic idea is to regard $D_k = \alpha_k^{-1} I$ as an approximation of the Hessian matrix $\bar{A}$ and then to impose some certain quasi-Newton property on the matrix D_k. More exactly, they minimize $\|D_k s_{k-1} - y_{k-1}\|_2$ where $s_{k-1} = x_k - x_{k-1}$ and $y_{k-1} = g_k - g_{k-1}$, yielding the following choice of α_k:

$$\alpha_k^{BB} = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T y_{k-1}} \ . \tag{1.8}$$

In the quadratic case, the above stepsize is equivalent to

$$\alpha_k = \frac{g_{k-1}^T g_{k-1}}{g_{k-1}^T \bar{A} g_{k-1}} \ , \tag{1.9}$$

which happens to be the Cauchy stepsize (1.7) at the previous iteration.

Although the Barzilai-Borwein (BB) stepsize (1.8) cannot guarantee a descent in the objective function or the gradient norm, the corresponding method is proved to be globally convergent for strictly convex quadratics (see Raydan [25]) and the convergence rate is R-linear (see Dai and Liao [13]). In the two-dimensional quadratic case, [3] presented a R-superlinear convergence result for the method. Dai and Fletcher [11] analyzed the asymptotic convergence behavior of the BB method for the higher-dimension case. In practical computations, it was pointed out in [3] that the BB stepsize (1.8) is far more efficient than the Cauchy stepsize (1.7). Fletcher [18] presented several linear systems of one million variables, showing that the BB method is comparable with the conjugate gradient method. The BB method has now received many generalizations and applications, for example see [26], [20], [5], [19], [11], [15], [23] and the references therein.

In this paper, we will apply the BB method to solve the saddle point problem (1.1). Each step of the Uzawa algorithm requires the solution of a symmetric positive definite linear system (see Step 2 of Algorithm 1.1). Elman and Golub [16] showed that this computation can be replaced by an approximate solution produced by an arbitrary iterative method, leading to the inexact Uzawa algorithm. The main purpose of this paper is to establish and analyze inexact BB algorithm for saddle point problems.

The rest of this paper is organized as follows. In the next section, we consider the inexact BB method where the exact gradient g_k is replaced by its some approximation $\tilde{g}_k$. Our study shows that there exists some small constant $\eta > 0$, which depends only on the problem dimension and the spectrum of the Hessian matrix, such that the BB method is R-linearly convergent for symmetric positive definite linear systems if $\|\tilde{g}_k - g_k\| \leq \eta \|\tilde{g}_k\|$ for all k. In Section 3, we propose the inexact BB algorithm for the saddle point problem (1.1). To establish the R-linear convergence result of this algorithm, we extend the result of Section 2 to the case when the previous gradient norm $\|\tilde{g}_{k-1}\|$ is used to control the inexactitude $\|\tilde{g}_k - g_k\|$. Although the estimate to η in our theoretical analyses can be very small, the numerical experiments in Section 4 show that this parameter η can be reasonably large without harming the convergence of the algorithm in practice. Further numerical results on some saddle point problems demonstrate the usefulness of the inexact BB algorithm. Conclusions and discussions are made in the last section.

2. Inexact Barzilai-Borwein Method

In this section we consider the problem of minimizing a strictly convex quadratic

$$\min \quad f(x) = \frac{1}{2}x^T Ax - b^T x, \tag{2.1}$$

where $A \in R^{n\times n}$ is symmetric positive definite and $b \in R^n$. To solve (2.1) we study the BB method with the gradient $g(x) = \nabla f(x) = Ax - b$ computed inexactly and call the method as *inexact BB method*. Assuming that $\tilde{g}_k$ is an approximation to g_k at the k-th iteration, the inexact BB algorithm for solving (2.1) can be described as follows.

Algorithm 2.1 *(Inexact BB)*

Step 1. Initialize $k = 0$ and pick some $x_0 \in \Re^n$. Calculate some approximation $\tilde{g}_0$ of the gradient $g_0 = \nabla f(x_0)$ and set $\tilde{\alpha}_0 = \tilde{g}_0^T\tilde{g}_0/\tilde{g}_0^T A\tilde{g}_0$;

Step 2. Update $x_{k+1} = x_k - \tilde{\alpha}_k\tilde{g}_k$ and $k = k+1$;

Step 3. Calculate some approximation $\tilde{g}_k$ of the gradient $g_k = \nabla f(x_k)$;

Step 4. Stop if some termination criterion is satisfied;

Step 5. Compute $s_{k-1} = x_k - x_{k-1}$, $\tilde{y}_{k-1} = \tilde{g}_k - \tilde{g}_{k-1}$ and $\tilde{\alpha}_k = \frac{s_{k-1}^T s_{k-1}}{s_{k-1}^T \tilde{y}_{k-1}}$, goto Step 2.

In the above algorithm, the first stepsize $\tilde{\alpha}_0$ is calculated by the steepest descent formula (1.7) with g_0 replaced by an inexact gradient $\tilde{g}_0$. This is not expensive if the matrix-vector product $A\tilde{g}_0$ can be used in computing $\tilde{g}_1$, as is the case of this paper.

Denote the error vector $\xi_k = \tilde{g}_k - g_k$. Then we have the basic relations

$$s_{k-1} = x_k - x_{k-1} = -\tilde{\alpha}_{k-1}\tilde{g}_{k-1}, \tag{2.2}$$

$$g_k = g_{k-1} - \tilde{\alpha}_{k-1}A\tilde{g}_{k-1}, \tag{2.3}$$

$$\tilde{g}_k = (I - \tilde{\alpha}_{k-1}A)\tilde{g}_{k-1} + \xi_k - \xi_{k-1}. \tag{2.4}$$

Further, still denoting $y_{k-1} = g_k - g_{k-1}$, we have that

$$\tilde{y}_{k-1} = y_{k-1} + \xi_k - \xi_{k-1}\,. \tag{2.5}$$

We are going to analyze Algorithm 2.1 under the condition that

$$\|\xi_k\| \le \eta \, \|\tilde{g}_k\| \,, \tag{2.6}$$

where $\eta \in (0,1)$ is some positive constant. It follows from (2.6) and the definition of ξ_k that

$$g_k^T \tilde{g}_k = \tilde{g}_k^T \tilde{g}_k + (g_k - \tilde{g}_k)^T \tilde{g}_k \ge \|\tilde{g}_k\|^2 - \|\xi_k\| \|\tilde{g}_k\| \ge (1-\eta)\|\tilde{g}_k\|^2.$$

Therefore we can see that the condition (2.6) ensures the descent property of $-\tilde{g}_k$ unless $\tilde{g}_k = 0$.

Suppose that the eigenvalues of the Hessian matrix A are

$$0 < \lambda_1 \le \lambda_2 \le ... \le \lambda_n. \tag{2.7}$$

The following theorem claims that Algorithm 2.1 is well defined if the parameter η satisfies

$$\eta \le \frac{1}{9} \left(\frac{\lambda_1}{\lambda_n} \right)^2 =: c_1. \tag{2.8}$$

Theorem 2.2 *Consider the inexact BB algorithm, namely, Algorithm 2.1 under the assumption (2.6). If η satisfies (2.8), the following relations*

$$\frac{2}{\lambda_1 + 2\lambda_n} \le \tilde{\alpha}_k \le \frac{2}{\lambda_1} \tag{2.9}$$

and

$$\|\tilde{g}_{k+1}\| \le c_2 \|\tilde{g}_k\|, \tag{2.10}$$

where $c_2 = \frac{9\,\lambda_n}{4\,\lambda_1} - 1$, hold for all $k \ge 0$. Therefore the algorithm is well defined.

Proof We establish (2.9) and (2.10) by induction. If $k = 0$, we have directly by the choice $\tilde{\alpha}_0 = \tilde{g}_0^T \tilde{g}_0 / \tilde{g}_0^T A \tilde{g}_0$ that $1/\lambda_n \le \tilde{\alpha}_0 \le 1/\lambda_1$. Thus (2.9) holds with $k = 0$. Further, fixing $k = 0$, we have by (2.9) that

$$\|I - \tilde{\alpha}_k \, A\| \le \max \left\{ 1 - \frac{2\lambda_1}{\lambda_1 + 2\lambda_n}, \; \frac{2\lambda_n}{\lambda_1} - 1 \right\} = \frac{2\lambda_n}{\lambda_1} - 1. \tag{2.11}$$

It follows from (2.4) that

$$\begin{aligned} \|\tilde{g}_{k+1}\| &\le \|(I - \tilde{\alpha}_k \, A) \, \tilde{g}_k\| + \|\xi_{k+1}\| + \|\xi_k\| \\ &\le \|I - \tilde{\alpha}_k \, A\| \, \|\tilde{g}_k\| + \eta \|\tilde{g}_{k+1}\| + \eta \|\tilde{g}_k\|. \end{aligned} \tag{2.12}$$

Using (2.12), (2.11) and noticing that $\eta \in (0, \frac{1}{9})$, we can obtain

$$\begin{aligned} \|\tilde{g}_{k+1}\| &\leq (1-\eta)^{-1}\left[\frac{2\lambda_n}{\lambda_1} - 1 + \eta\right] \|\tilde{g}_k\| \\ &\leq \left[\frac{2\lambda_n}{\lambda_1}(1-\eta)^{-1} - 1\right] \|\tilde{g}_k\| \\ &\leq \left[\frac{9\lambda_n}{4\lambda_1} - 1\right] \|\tilde{g}_k\|. \end{aligned} \tag{2.13}$$

Thus by the choice of c_2, we know that (2.10) is true with $k = 0$.

Now we assume that (2.9) and (2.10) hold for all $l \leq k$, where k is some integer that satisfies $k \geq 0$. Then by (2.5), (2.6), the induction assumption and (2.8), we have the following estimate

$$\begin{aligned} \left|\frac{s_k^T \tilde{y}_k}{s_k^T s_k} - \frac{s_k^T y_k}{s_k^T s_k}\right| &\leq \left|\frac{s_k^T(\xi_{k+1} - \xi_k)}{s_k^T s_k}\right| \leq \frac{\|\xi_{k+1}\| + \|\xi_k\|}{\|s_k\|} \\ &\leq \frac{(1+c_2)\eta\|\tilde{g}_k\|}{\tilde{\alpha}_k\|\tilde{g}_k\|} = \frac{9\lambda_n\eta}{4\lambda_1\tilde{\alpha}_k} \\ &\leq \frac{9\lambda_n}{4\lambda_1} \cdot \frac{\lambda_1 + 2\lambda_n}{2} \cdot \frac{1}{9}\left(\frac{\lambda_1}{\lambda_n}\right)^2 < \frac{\lambda_1}{2}\,. \end{aligned} \tag{2.14}$$

On the other hand, we can see from the definition of y_k that $\lambda_1 \leq \frac{s_k^T y_k}{s_k^T s_k} \leq \lambda_n$. It follows from this and (2.14) that

$$\frac{\lambda_1}{2} \leq \left|\frac{s_k^T \tilde{y}_k}{s_k^T s_k}\right| \leq \lambda_n + \frac{\lambda_1}{2}. \tag{2.15}$$

Thus $\tilde{\alpha}_{k+1}$ is well defined and (2.9) holds with k replaced by $k+1$. Further, it is not difficult to see that the deductions from (2.11) to (2.13) are still available and hence (2.10) is true when the index k is replaced with $k+1$. Therefore by induction, the relations (2.9) and (2.10) hold for all $k \geq 0$. Algorithm 2.1 is then well defined. *q.e.d*

In the case that (2.6) and (2.8) hold, we can see from Theorem 2.2 that the inexact BB algorithm is well defined provided that the initial stepsize α_0 satisfies the relation (2.9). Theorem 2.2 also tells us that to guarantee the well-definition of the algorithm, the constant η need be less than the square of the inverse of the condition number λ_n/λ_1 of the matrix A. The following theorem extends the R-linear convergence result in [13] of the BB method.

Theorem 2.3 *Consider the inexact BB algorithm, namely, Algorithm 2.1 under the conditions (2.6) and (2.8). There exists some positive constant* $c_3 \leq c_1$*, which depends only on* λ_1*,* λ_n *and the dimension* n*, such that if* $\eta \leq c_3$*, the algorithm either gives the solution in finite iterations or converges to the solution R-linearly.*

Proof To prove the theorem, we compare the inexact BB algorithm and the (exact) BB method. At first, notice that the (exact) BB iterations are uniquely decided by its starting point and initial stepsize. By Lemma 2.4 in [13] and some slight modifications, we can see that there exists some positive integer M, which depends only λ_1, λ_n and the problem dimension n, such that for any starting point $z_0 \in \Re^n$ and initial stepsize β_0 satisfying $c_L \leq \beta_0 \leq c_U$ (c_L and c_U are some fixed positive constants), the M-th point z_M generated by the (exact) BB method satisfies

$$\|\nabla f(z_M)\| \leq \frac{1}{2} \|\nabla f(z_0)\|. \tag{2.16}$$

Let's fix $c_L = 2/(\lambda_1 + 2\lambda_n)$ and $c_U = 2/\lambda_1$ and take some M satisfying the above statement. For any point x_k and stepsize $\tilde{\alpha}_k$ generated by Algorithm 2.1, we consider the (exact) BB iterations $\{z_{k+l};\ l \geq 0\}$ with $z_k = x_k$ and $\beta_k = \tilde{\alpha}_k$. The gradient of f at z_{k+l} is denoted by h_{k+l}. It is easy to see that $h_k = g_k$ and for $l \geq 0$,

$$h_{k+l+1} = (I - \beta_{k+l}\, A) h_{k+l} \quad \text{and} \quad \beta_{k+l+1} = \frac{h_{k+l}^T h_{k+l}}{h_{k+l}^T A h_{k+l}}\,. \tag{2.17}$$

Further, we take the smallest integer $m' \leq M$ such that $\|h_{k+m'}\| \leq \frac{1}{2}\|h_k\|$. In this case, we have that

$$\|h_{k+l}\| > \frac{1}{2}\|h_k\|, \quad \text{for all } l = 0, 1, \ldots, m'-1. \tag{2.18}$$

We now consider the quantities $\phi_{k+l} = \|\tilde{g}_{k+l} - h_{k+l}\|$ and $\psi_{k+l} = |\tilde{\alpha}_{k+l} - \beta_{k+l}|$. Using (2.4), (2.17), (2.6) and (2.10), we get that for $l \geq 0$

$$\begin{aligned} \phi_{k+l+1} &\leq \|(I - \beta_{k+l}A) h_{k+l} - (I - \tilde{\alpha}_{k+1}) A \tilde{g}_{k+l}\| + \|\xi_{k+l+1}\| + \|\xi_{k+l}\| \\ &\leq \|I - \beta_{k+l}A\|\, \phi_{k+l} + \|A\|\, \|\tilde{g}_{k+l}\|\, \psi_{k+l} + \eta(1 + c_2)\|\tilde{g}_{k+l}\|\,, \\ &\leq (\frac{\lambda_n}{\lambda_1} - 1)\phi_{k+l} + \lambda_n \|\tilde{g}_{k+l}\|\, \psi_{k+l} + \eta(1 + c_2)\|\tilde{g}_{k+l}\|\,. \end{aligned} \tag{2.19}$$

On the other hand, we have by direct calculations that

$$\psi_{k+l+1} = \left| \frac{\tau_{k+l} + \|h_{k+l}\|^2 \, \theta_{k+l}}{h_{k+l}^T A h_{k+l} \, [\, \tilde{g}_{k+l}^T A \tilde{g}_{k+l} + \theta_{k+l} \,]} \right| , \tag{2.20}$$

where

$$\tau_{k+l} = \|h_{k+l}\|^2 \, \tilde{g}_{k+l}^T A \tilde{g}_{k+l} - h_{k+l}^T A h_{k+l} \, \|\tilde{g}_{k+l}\|^2 \tag{2.21}$$

and

$$\theta_{k+l} = \tilde{\alpha}_{k+l}^{-1} \tilde{g}_{k+l}^T (\xi_{k+l} - \xi_{k+l+1}) \, . \tag{2.22}$$

For τ_{k+l}, we have the estimate

$$\begin{aligned}
|\tau_{k+l}| &\le \left| \tilde{g}_{k+l}^T A \tilde{g}_{k+l} [\|h_{k+l}\|^2 - \|\tilde{g}_{k+l}\|^2] + \|\tilde{g}_{k+l}\|^2 [\tilde{g}_{k+l}^T A \tilde{g}_{k+l} - h_{k+l}^T A h_{k+l}] \right| \\
&\le \left[\tilde{g}_{k+l}^T A \tilde{g}_{k+l} (\|h_{k+l}\| + \|\tilde{g}_{k+l}\|) + \|\tilde{g}_{k+l}\|^2 \, \|A(h_{k+l} + \tilde{g}_{k+l})\| \right] \phi_{k+l} \\
&\le 2\lambda_n \|\tilde{g}_{k+l}\|^2 (\|h_{k+l}\| + \|\tilde{g}_{k+l}\|) \phi_{k+l}.
\end{aligned} \tag{2.23}$$

For θ_{k+l}, we have the estimate

$$|\theta_{k+l}| \le \frac{(1+c_2)(\lambda_1 + 2\lambda_n)}{2} \eta \|\tilde{g}_{k+l}\|^2 \le \frac{\lambda_1}{2} \|\tilde{g}_{k+l}\|^2. \tag{2.24}$$

Denote the constant $c_4 = \frac{(1+c_2)(\lambda_1+2\lambda_n)}{\lambda_1}$. Using (2.21) and (2.24), we can get from (2.20) that

$$\begin{aligned}
\psi_{k+l+1} &\le \frac{2\lambda_n \|\tilde{g}_{k+l}\|^2 (\|h_{k+l}\| + \|\tilde{g}_{k+l}\|) \, \phi_{k+l} + \frac{c_4 \lambda_1}{2} \eta \|h_{k+l}\|^2 \|\tilde{g}_{k+l}\|^2}{h_{k+l}^T A h_{k+l} \, [\tilde{g}_{k+l}^T A \tilde{g}_{k+l} - \frac{\lambda_1}{2} \|\tilde{g}_{k+l}\|^2]} \\
&\le \frac{2\lambda_n (\|h_{k+l}\| + \|\tilde{g}_{k+l}\|) \, \phi_{k+l} + \frac{c_4 \lambda_1}{2} \eta \|h_{k+l}\|^2}{\frac{\lambda_1}{2} \|h_{k+l}\|^2} \\
&\le \frac{4\lambda_n (\|\tilde{g}_{k+l}\| + \|h_{k+l}\|) \, \phi_{k+l}}{\lambda_1 \|h_{k+l}\|^2} + c_4 \eta \, .
\end{aligned} \tag{2.25}$$

It follows from (2.10) that $\|\tilde{g}_{k+l}\| \le c_2^l \|\tilde{g}_k\|$. By (2.17), we can deduce that $\|h_{k+l}\| \le c_5^l \|h_k\|$, where $c_5 = \max\{1, \frac{\lambda_n}{\lambda_1} - 1\}$ is constant. Since $h_k = g_k$, we have from (2.6) and $\eta \le c_1 < \frac{1}{9}$ that $\|\tilde{g}_k\| \le \frac{9}{8} \|h_k\|$. Using these analyses, the relation (2.18), (2.25) and $m' \le M$, we can obtain the estimate

$$\psi_{k+l+1} \|\tilde{g}_{k+l+1}\| \le c_6 \phi_{k+l} + c_4 \eta \|\tilde{g}_{k+l+1}\|, \quad \text{for } l = 0, 1, \ldots, m' - 2, \tag{2.26}$$

where c_6 is the constant given by

$$c_6 = \frac{9\lambda_n(9c_2^{M-2} + 8c_5^{M-2})c_2^{M-1}}{4\lambda_1}\,. \tag{2.27}$$

Since the first stepsize β_k is chosen to be $\tilde{\alpha}_k$, we have that $\psi_k = 0$. In addition, (2.6) implies that $\phi_k \le \eta\|\tilde{g}_k\|$. Thus we have from (2.19) that

$$\phi_{k+1} \le (\frac{\lambda_n}{\lambda_1} - 1)\phi_k + \eta(1 + c_2)\|\tilde{g}_k\| \le (c_2 + \frac{\lambda_n}{\lambda_1})\eta\|\tilde{g}_k\|\,. \tag{2.28}$$

For $l \in [1, m' - 1]$, we have by (2.19) and (2.26) (with l replaced by $l - 1$) and $\|\tilde{g}_{k+l}\| \le c_2^M\|\tilde{g}_k\|$ that

$$\phi_{k+l+1} \le (\frac{\lambda_n}{\lambda_1} - 1)\phi_{k+l} + c_6\lambda_n\phi_{k+l-1} + (1 + c_2 + c_4\lambda_n)\eta\|\tilde{g}_{k+l}\|\,. \tag{2.29}$$

Notice that all the constants c_1, c_2, c_4, c_5, c_6 and the integer M are dependent only on λ_1, λ_n and possibly the dimension n. From $\phi_k \le \eta\|\tilde{g}_k\|$, (2.28), (2.29) and $\|\tilde{g}_{k+l}\| \le c_2^M\|\tilde{g}_k\|$, we know that there exists some constant c_7, which depends only on λ_1, λ_n and the dimension n, such that

$$\phi_{k+m'} \le c_7\,\eta\,\|\tilde{g}_k\|. \tag{2.30}$$

Taking $c_3 = \min\{\frac{1}{3+6c_7},\, c_1\}$, we obtain from (2.30), $\|h_{k+m'}\| \le \frac{1}{2}\|h_k\|$, $\|h_k\| \le (1+\eta)\|\tilde{g}_k\|$ and $\eta \le c_3$ that

$$\|\tilde{g}_{k+m'}\| \le \|h_{k+m'}\| + \phi_{k+m'} \le \left[\frac{(1+\eta)}{2} + c_7\eta\right]\|\tilde{g}_k\| \le \frac{2}{3}\,\|\tilde{g}_k\|\,. \tag{2.31}$$

To complete the proof, we define a subsequence $\{k_i\}$ with $k_1 = 2$ for Algorithm 2.1. If k_i has been decided, we choose $k_{i+1} = k_i + m_i$, where $m_i \in [1,\, m]$ is so chosen that

$$\|\tilde{g}_{k_{i+1}}\| \le \frac{2}{3}\,\|\tilde{g}_{k_i}\|. \tag{2.32}$$

By the analysis in the previous paragraph, we know that this is possible. It then follows that

$$\|\tilde{g}_{k_i}\| \le (\frac{2}{3})^{i-1}\|\tilde{g}_{k_1}\| \tag{2.33}$$

with $k_i = k_1 + \sum_{j=1}^{i-1} m_j \le k_1 + M(i-1)$. Consequently, we have that

$$\limsup_{i\to\infty}\,\|\tilde{g}_{k_i}\|^{\frac{1}{k_i}} \le (\frac{2}{3})^{\frac{1}{M}}. \tag{2.34}$$

From the above relation and (2.10), we know that $\|\tilde{g}_k\|$ and hence $\|g_k\|$ converges to zero R-linearly. *q.e.d*

The importance of Theorems 2.2 and 2.3 is in that, to guarantee the well-definition of Algorithm 2.1 and inherent the R-linear convergence of the (exact) BB method, the calculation error of the gradient can be less than some constant proportion of the gradient norm. The constant depends only the dimension n and the minimal and maximal eigenvalues of the matrix A. However, the current estimate to the constant η in the proof of Theorem 2.3 may be very small, since the integer M can be very large. In practice, the choice of the value η is optimistic, as will be seen in our numerical experiments of Section 4.

As one application of Theorem 2.3, we can show that the (exact) BB method is locally R-linearly convergent for twice continuously differentiable functions. Suppose that $f(x)$ is the function to be minimized and x^* is a point at which $\nabla f(x^*)$ and its Hessian H^* is positive definite. Then at some neighbourhood of x^*, the (exact) BB method for the minization of $f(x)$ can be regarded as the inexact BB method for minimizing the following quadratic

$$q(x) = f(x^*) + \frac{1}{2}(x - x^*)^T H^*(x - x^*). \tag{2.35}$$

In the case that f is twice continuously differentiable, it is not difficult to establish the relation

$$\|\nabla f(x) - \nabla q(x)\| = o(\|\nabla f(x)\|). \tag{2.36}$$

Hence the condition (2.6) must be satisfied when x_k tends to x^* and hence R-linear convergence can be established. This remark weakens the assumption that the objection function f is two times Lipschitz continuously differentiable for the CBB method in [12], in which case the following relation holds

$$\|\nabla f(x) - \nabla q(x)\| = O(\|\nabla f(x)\|^2). \tag{2.37}$$

3. Inexact BB method for saddle point problems

The Uzawa algorithm for the saddle point problem (1.1) requires the solution of a linear system at each iteration (see Step 2 of Algorithm 1.1). In practice, it is usually expensive to solve the subproblem exactly. Elman and Golub [16] proposed to replace Step 2 of Algorithm 1.1 by

$$Au_{k+1} = f - B^T p_k + \delta_k, \tag{3.1}$$

where the vector δ_k is the residual of the approximation solution u_{k+1} to the system $Av = f - B^T p_k$. They suggested that a natural choice for the magnitude of δ_k is

$$\|\delta_k\| \leq \tau \|Cp_{k-1} - Bu_k + h\| \,. \tag{3.2}$$

This is because the quantity $Cp_{k-1} - Bu_k + h$ is the residual of the second block row of (1.1) for the approximation solution pair (u_k, p_{k-1}) and this quantity has already been calculated for the update of p_k in the previous step.

If the subproblem at Step 1 of Algorithm 1.1 is exactly solved, we obtain the exact residual to the system (1.3):

$$g_k = \bar{A}p_k - \bar{b} = Cp_k - BA^{-1}(f - B^T p_k) + h. \tag{3.3}$$

When the subproblem is solved inexactly by (3.1), the inexact gradient $\tilde{g}_k$ can be written as

$$\tilde{g}_k = Cp_k - Bu_{k+1} + h = g_k - BA^{-1}\delta_k. \tag{3.4}$$

From (3.2) and the first equality of (3.4) with k replaced by $k-1$, the error vector δ_k is required to satisfy

$$\|\delta_k\| \leq \tau \, \|\tilde{g}_{k-1}\|. \tag{3.5}$$

Combining the BB method and the inexact idea of Golub and Elman, we give an inexact BB algorithm for saddle point problem (1.1).

Algorithm 3.1 *(Inexact BB algorithm for saddle point problems)*

Step 1. Initialize $k = 0$ *and* $p_0 \in \Re^m$*. Choose some big constant* $\rho > 1$ *and some initial stepsize* $\tilde{\alpha}_0 \in [\rho^{-1}, \, \rho]$;

Step 2. Compute u_{k+1} *such that* $Au_{k+1} = f - B^T p_k + \delta_k$ *with* δ_k *satisfying (3.2);*

Step 3. Compute $\tilde{g}_k = Cp_k - Bu_{k+1} + h$, $p_{k+1} = p_k - \tilde{\alpha}_k \tilde{g}_k$ *and set* $k = k+1$*;*

Step 4. Stop if some termination criterion is satisfied;

Step 5. Calculate $s_{k-1} = p_k - p_{k-1}$ *and* $\tilde{y}_{k-1} = \tilde{g}_k - \tilde{g}_{k-1}$*. Compute the next stepsize* $\tilde{\alpha}_k$ *by* $\left[\max\{\rho^{-1}, \, \min\{\frac{s_{k-1}^T \tilde{y}_{k-1}}{s_{k-1}^T s_{k-1}}, \, \rho\}\}\right]^{-1}$ *and goto Step 2.*

The introduction of the constant $\rho > 1$ ensures that the stepsize $\tilde{\alpha}_k$ and hence Algorithm 3.1 is well defined. The previous section considers the inexact BB algorithm under the assumption (2.6). For Algorithm 3.1, however, we have by (3.4) and (3.5) that

$$\|\xi_k\| = \|\tilde{g}_k - g_k\| \leq \bar{\eta}\, \|\tilde{g}_{k-1}\|, \tag{3.6}$$

where $\bar{\eta} = \tau \|BA^{-1}\|$. Since it is possible that $\|g_k\|$ can be arbitrarily smaller than $\|g_{k-1}\|$ in the (exact) BB method and hence it is likely that $\|\tilde{g}_k\|$ is far smaller than $\|\tilde{g}_{k-1}\|$, (3.6) does not imply (2.6) for any small constant τ. Therefore we cannot establish the R-linear convergence of Algorithm 3.1 directly from Theorem 2.3. At the same time, we can see that unlike (2.6), the condition (3.6) cannot ensure the descent property of $-\tilde{g}_k$ at every iteration.

The above difficulties can be circumvented by noticing that if $\|\tilde{g}_k\|$ is significantly less than $\|\tilde{g}_{k-1}\|$, then a good approximation $\tilde{g}_k$ of the gradient g_k has been obtained. On the other hand, it follows from (2.4), $\tilde{\alpha}_k \in [\rho^{-1}, \rho]$ and (3.6) that

$$\|\tilde{g}_k\| \leq \lambda_n \, \rho \|\tilde{g}_{k-1}\| + \bar{\eta} \|\tilde{g}_{k-2}\|. \tag{3.7}$$

Here and below we assume that ρ is a very large constant such that

$$\rho \geq \max\{\frac{2}{\lambda_1},\, \lambda_n + \frac{\lambda_1}{2}\}. \tag{3.8}$$

The above relation and $\lambda_1 \leq \lambda_n$ implies that $\lambda_n \rho \geq 2$. From (3.6), we have

$$\|g_k\| \leq \|\tilde{g}_k\| + \bar{\eta}\, \|\tilde{g}_{k-1}\|. \tag{3.9}$$

The relations (3.7) and (3.9) hint us that, to establish the R-linear convergence of Algorithm 3.2, we need to consider a subsequence of k_i such that the approximation gradients $\tilde{g}_{k_i-1}$ and $\tilde{g}_{k_i}$ have some properties simultaneously.

Theorem 3.2 *Consider Algorithm 3.1 for saddle point problem (1.1) under the conditions (2.6) and (2.8). Assume that ρ is a big constant that satisfies (3.8). Then there exists some positive constant τ_1, which depends only on λ_1, λ_n and the dimension n, such that if (3.5) holds for all k and $\tau \leq \tau_1$, the algorithm either gives the solution in finite iterations or converges to the solution R-linearly.*

Proof Similarly to the proof of Theorem 2.3, we find some constant $c_3 \in (0, 1)$ and integer m which depend only on λ_1, λ_n, the dimension n and the parameter ρ such that, if $\|\xi_j\| \leq \eta\|\tilde{g}_j\|$ for all j and if $\eta \leq c_3$, then for any index k, there exists some integer $m' \leq M$ satisfying $\|\tilde{g}_{k+m'}\| \leq \frac{2}{3}\|\tilde{g}_k\|$. Denote $M_1 = \lceil \frac{\log(2\lambda_n\rho)}{\log 1.5} \rceil M$. Then if $\|\xi_j\| \leq \eta\|\tilde{g}_j\|$ for all j and if $\eta \leq c_3$, then for any index k, there exists some integer $m' \leq M_1$ such that

$$\|\tilde{g}_{k+m'}\| \leq (\frac{2}{3})^{\frac{M_1}{M}}\|\tilde{g}_k\| \leq \frac{1}{2\lambda_n\rho}\|\tilde{g}_k\|. \tag{3.10}$$

Now we denote the constants

$$c_8 = \frac{1}{2\lambda_n\rho(\lambda_n\rho+1)^{M_1-1}}, \quad \tau_1 = c_3c_8\|BA^{-1}\|^{-1}. \tag{3.11}$$

and define a subsequence $\{k_i\}$ in the following way. Pick the least index $k_1 \geq 1$ such that $\|\tilde{g}_{k_1}\| \geq \frac{2}{3}\|\tilde{g}_{k_1-1}\|$. If this is not possible, we have $\|\tilde{g}_k\| \leq \frac{2}{3}\|\tilde{g}_{k-1}\|$ for all $k \geq 1$ and $\{\|\tilde{g}_k\|\}$ is a Q-linear convergence sequence. Assume that for some $i \geq 1$, k_i has been chosen with the property

$$\|\tilde{g}_{k_i}\| \geq \frac{2}{3}\|\tilde{g}_{k_i-1}\|. \tag{3.12}$$

By this, (3.9) and $\bar{\eta} \leq \frac{2}{3}$ and using the induction principle, it is not difficult to show that

$$\|\tilde{g}_{k_i+l}\| \leq (\lambda_n\rho+1)^l\|\tilde{g}_{k_i}\|, \quad \text{for all } l \geq 1. \tag{3.13}$$

Thus if $\|\tilde{g}_{k_i+l}\| < c_8\|\tilde{g}_{k_i+l-1}\|$ for some $l \in [1, M_1]$, we have by this, (3.13) and the choice of c_8 that $\|\tilde{g}_{k_i+l}\| \leq \frac{1}{2\lambda_n\rho}\|\tilde{g}_{k_i}\|$. If this is not the case, we have $\|\tilde{g}_{k_i+l}\| \geq c_8\|\tilde{g}_{k_i+l-1}\|$ for $l = 1, \ldots, M_1$. It follows from this, (3.6) and (3.11) that $\|\xi_{k_i+l}\| \leq c_3\|\tilde{g}_{k_i+l}\|$ for $l = 1, \ldots, M_1$. By the statement in the first paragraph of the proof and the choice of $\bar{\eta}$, we must have that $\|\tilde{g}_{k_i+M_1}\| \leq \frac{1}{2\lambda_n\rho}\|\tilde{g}_{k_i}\|$. Therefore there always exists some integer $m' \in [1, M_1]$ such that

$$\|\tilde{g}_{k_i+m'}\| \leq \frac{1}{2\lambda_n\rho}\|\tilde{g}_{k_i}\|. \tag{3.14}$$

We then take k_{i+1} to be the least integer that is not less than $k_i + m' + 1$ and satisfies $\|\tilde{g}_{k_{i+1}}\| \geq \frac{2}{3}\|\tilde{g}_{k_{i+1}-1}\|$. If $k_{i+1} = +\infty$, we know that $\{\|\tilde{g}_k\|\}$ is R-linearly convergent. Therefore we assume that the above-defined $\{k_i\}$ is an infinite sequence.

If $k_{i+1} = k_i + m' + 1$, we have by (3.7), (3.14), (3.13), (3.11), $m' \le M_1$ and the definition of $\bar\eta$ that

$$\begin{aligned}
\|\tilde g_{k_{i+1}}\| &\le \lambda_n\,\rho\,\|\tilde g_{k_i+m'}\| + \bar\eta\,\|\tilde g_{k_i+m'-1}\| \\
&\le \lambda_n\rho\left(\frac{1}{2\lambda_n\rho}\|\tilde g_{k_i}\|\right) + \bar\eta\,(\lambda_n\rho+1)^{m'-1}\,\|\tilde g_{k_i}\| \\
&\le \left[\frac{1}{2} + \frac{c_3}{2\lambda_n\rho}\right]\|\tilde g_{k_i}\| \;\le\; \frac{3}{4}\,\|\tilde g_{k_i}\|\;. \qquad (3.15)
\end{aligned}$$

The facts that $c_3 \in (0,\,1)$ and $\lambda_n\rho \ge 2$ are also used for the last inequality.

If $k_{i+1} > k_i + m' + 1$, we denote $j_0 = k_{i+1} - (k_i + m' + 1)$. The choice of k_{i+1} implies that

$$\|\tilde g_{k_i+m'+j}\| \le (\,\frac{2}{3}\,)^j\|\tilde g_{k_i+m'}\|, \quad \text{for } j = 1,\ldots,j_0\;. \qquad (3.16)$$

It follows by (3.7), (3.16), (3.11) and the fact that $3\bar\eta \le \lambda_n\rho$ that

$$\begin{aligned}
\|\tilde g_{k_{i+1}}\| &\le \lambda_n\,\rho\,\|\tilde g_{k_i+m'+j_0}\| + \bar\eta\,\|\tilde g_{k_i+m'+j_0-1}\| \\
&\le (\,\frac{2}{3}\,)^{j_0}\lambda_n\,\rho\,\|\tilde g_{k_i+m'}\| + \bar\eta\,(\,\frac{2}{3}\,)^{j_0-1}\|\tilde g_{k_i+m'}\| \\
&\le (\,\frac{2}{3}\,)^{j_0}\left(\frac{1}{2}\,\|\tilde g_{k_i}\|\right) + \bar\eta\,(\,\frac{2}{3}\,)^{j_0-1}\left(\frac{1}{2\lambda_n\rho}\,\|\tilde g_{k_i}\|\right) \\
&\le (\,\frac{2}{3}\,)^{j_0}\left[\frac{1}{2} + \frac{3\bar\eta}{4\lambda_n\rho}\right]\|\tilde g_{k_i}\| \;\le\; (\,\frac{2}{3}\,)^{j_0}\left[\frac{3}{4}\,\|\tilde g_{k_i}\|\right]\;. \qquad (3.17)
\end{aligned}$$

Combining the two possible cases of k_{i+1} and noting that $m' \le M_1$, we always have the following relation

$$\|\tilde g_{k_{i+1}}\| \le \frac{3}{4}\,(\,\frac{2}{3}\,)^{\max\{0,k_{i+1}-k_i-M_1-1\}}\,\|\tilde g_{k_i}\|\;. \qquad (3.18)$$

Denote the constant $c_9 = (\,\frac{3}{4}\,)^{\frac{1}{M_1+1}}$. Since $M_1 \ge 1$, we have that $\frac{2}{3} < c_9 < 1$. With the choice of c_9, we can show by (3.18) that $\|\tilde g_{k_{i+1}}\| \le c_9^{k_{i+1}-k_i}\,\|\tilde g_{k_i}\|$. The recursion of this relation leads to

$$\|\tilde g_{k_{i+1}}\| \le c_9^{k_{i+1}-k_1}\,\|\tilde g_{k_1}\|\;. \qquad (3.19)$$

In addition, by (3.13), the definition of k_{i+1} and $m' \le M_1$, it is not difficult to see that

$$\max\{\|\tilde g_{k_i+1}\|,\ldots,\|\tilde g_{k_{i+1}-1}\|\} \le (\lambda_n\rho+1)^{M_1-1}\|\tilde g_{k_i}\|\;. \qquad (3.20)$$

Therefore we know by (3.19) and (3.20) that

$$\limsup_{k\to\infty} \|\tilde{g}_k\|^{\frac{1}{k}} \le c_9 < 1, \tag{3.21}$$

which implies that $\{\|\tilde{g}_k\|\}$ and hence by (3.9), $\{\|g_k\|\}$ are R-linearly convergent. *q.e.d*

Again, the proof to Theorem 3.2 provides a pessimistic estimate to the largest admissible value to τ_1. Nevertheless, the numerical experiments in the next section show that τ_1 and hence $\bar{\eta}$ can be much larger.

4. Numerical Experiments

Our numerical experiments in this section are divided into two parts. In the first part, we test Algorithm 2.1 with the inexactitude case (3.6) for random symmetric and positive definite linear systems. Specifically, we observe how the value of $\bar{\eta}$ in (3.6) influences the performance of the algorithm and how its choice depends on the problem dimension n and the smallest eigenvalue λ_1 and the condition number κ of the matrix A.

Problem 4.1. *Consider the symmetric positive definite linear system $Ax = b$, where $x \in \Re^{n\times n}$. The coefficient matrix A is formed by*

$$A = PDP^T, \quad \textit{where } P = (1 - 2\omega_1\omega_1^T)(1 - 2\omega_2\omega_2^T)(1 - 2\omega_3\omega_3^T) \tag{4.1}$$

and ω_1, ω_2 and ω_3 are unit vectors generated by the uniform distribution in $\Re^n$ and D is a diagonal matrix. Given the smallest eigenvalue λ_1 and the condition number κ, the i-th diagonal entry $D_{i,i}$ of the matrix D is set to

$$D_{i,i} = \exp\left(\log\lambda_1 + \frac{i-1}{n-1}\log\kappa\right). \tag{4.2}$$

To generate the right hand term b, we randomly generate a solution $x^* \in \Re^n$ with $x_i^* \in [-1, 1]$. Then we set $b = Ax^*$. The starting point is $x_0 = 0$.

Given $t_{\min}$, $t_{\max}$ and some positive integer $n_{\bar{\eta}}$, we tested the inexact BB method with the inexactitude case (3.6) using the following values for $\bar{\eta}$:

$$\bar{\eta}_j = \exp\left(\log t_{\min} + \sqrt{(j-1)/(n_{\bar{\eta}}-1)}\,\log(t_{\max}/t_{\min})\right), \quad j = 1, \ldots, n_{\bar{\eta}}.$$

Since $\|\tilde{g}_{-1}\|$ is not available for the first iteration, we assumed that $\|\tilde{g}_{-1}\| = 0$, which means that the exact gradient g_0 is used. The initial stepsize is set to be the Cauchy stepsize $\alpha_0 = g_0^T g_0 / g_0^T A g_0$ as in Step 1 of Algorithm 2.1. For $k \geq 1$, to generate an approximation $\tilde{g}_k$ of the exact gradient g_k, we first generate a random vector v_k and then set

$$\tilde{g}_k = g_k + \bar{\eta} \, \frac{\|\tilde{g}_{k-1}\|}{\|v_k\|} \, v_k.$$

The above choice of $\tilde{g}_k$ is such that the equality in the relation (3.6) holds. Although the condition (3.6) cannot guarantee the descent property of $-\tilde{g}_k$, Theorem 3.2 tells us that the inexact BB method still converges and the convergence rate is R-linear.

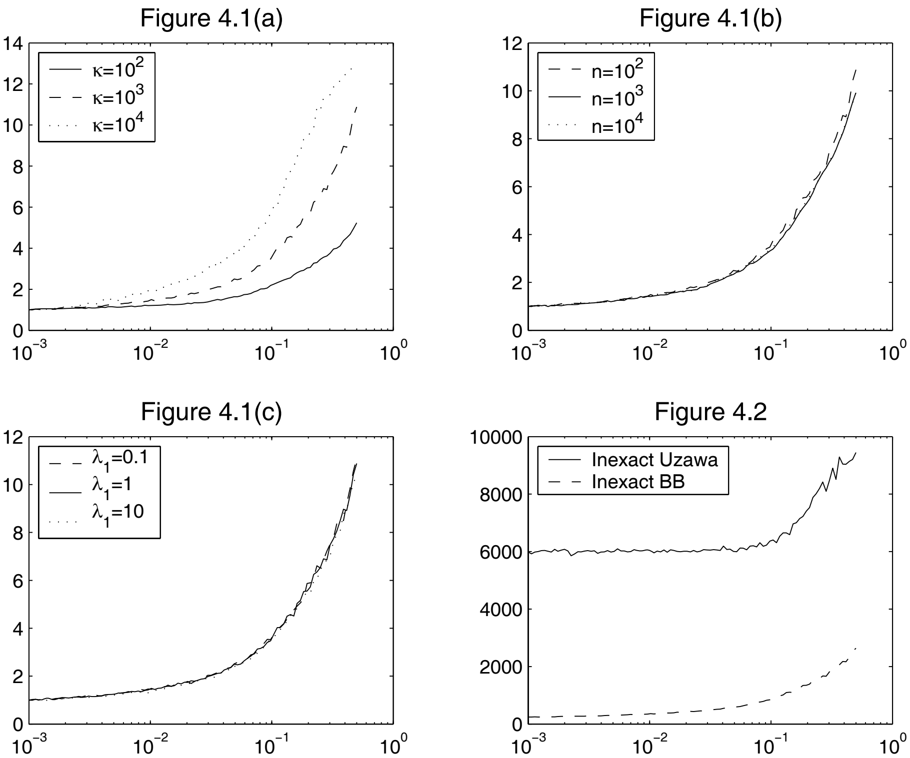
In our tests, we fix $t_{\min} = 10^{-3}$, $t_{\max} = 0.5$ and $n_{\bar{\eta}} = 100$ and observe the influence of κ, n and λ_1. For each value of κ, n and λ_1, we do 100 tests and use the stopping condition

$$\|g_k\| \leq 10^{-6} \, \|g_0\|, \tag{4.3}$$

where the exact gradient is used for the purpose of comparison. All tests for this part were done with MATLAB 6.5.0.

At first, to observe the influence of κ, we fix $n = 100$ and $\lambda_1 = 1$ and use the following three values for κ: 10^2, 10^3 and 10^4. For $j = 1, \ldots, n_{\bar{\eta}}$, we denote by $Iter(j)$ the average iteration numbers of the 100 tests required for the inexactitude rule (3.6) with $\bar{\eta} = \bar{\eta}_j$. For the three different values of κ, Figure 4.1(a) plots the corresponding curves $Iter(j)/Iter(1)$ vs $\bar{\eta}_j$. Secondly, we fix $\kappa = 10^3$ and $\lambda_1 = 1$ and vary the problem dimension n to be 10^2, 10^3 and 10^4, respectively. Thirdly, we fix $\kappa = 10^3$ and $n = 10^2$ and vary the minimal eigenvalue λ_1 to be 0.1, 1 and 10, respectively. See Figures 4.1(b) and 4.1(c) for the corresponding curves. Considering the log scale of the horizontal axis, we can see from the three figures that basically, as the inexactness parameter $\bar{\eta}$ increases, the required average iteration number is linearly increases and hence the R-linear factor of the inexact BB algorithm is linearly decreasing. The decrement of the R-linear factor is strongly affected by the condition number of the problem, but has little relation with the problem dimension or the minimal eigenvalue of the coefficient matrix A. In the numerical sense, this feature resembles to that of the inexact Uzawa algorithm, whose Q-linear factor mainly depends on κ and $\bar{\eta}$ (see Theorem 2.2 in [16]).

A comparison was also made between the inexact BB algorithm and the inexact Uzawa algorithm. In this case, we fix $n = 100$, $\kappa = 10^3$ and $\lambda_1 = 1$.



(Note: all the x axes in the above figures are corresponding to the value of $\bar{\eta} = \eta_j$. The y axes in Figures 4.1 (a) to (c) are corresponding to the relative value of $Iter(j)$ with respect to $Iter(1)$; the y axis in Figure 4.2 stand for the absolute value of $Iter(j)$.)

Therefore the only difference between the two algorithms is the choice of the stepsize α_k. The inexact Uzawa algorithm uses (1.5), which means that $\alpha_k \equiv 2/101$, whereas the inexact BB algorithm decides the stepsize according to the information at the most recent two points except the initial stepsize is chosen to be the Cauchy stepsize. In Figure 4.2, we plot the curves of the required average iteration numbers $Iter(j)$ vs $\bar{\eta}_j$. From the figure, we can see that the inexact BB algorithm is far more efficient than the inexact Uzawa algorithm. In the case when $\bar{\eta} = \bar{\eta}_1 = 10^{-3}$, to reach the stopping condition (4.3), the inexact BB algorithm and the inexact Uzawa algorithm requires 242.6 and 5983.5 iterations on the average. From Figure 4.2, we can also see that the influence of the inexactness parameter $\bar{\eta}$ is similar to the performance of the two algorithms.

In the second part of our numerical experiments we test Algorithm 3.1 on saddle point problems arising from the finite element discretization of Stokes equations.

Problem 4.2. *This problem is related to the Stokes equations:*

$$\begin{array}{rl} -\triangle u + \nabla p = f & in\ \Omega = (0,1)\times(0,1) \\ -div\, u = h & in\ \Omega \\ u = 0 & on\ \partial\Omega \\ \int_\Omega p = 0 & \end{array} \tag{4.4}$$

We discretize (4.4) in the same way as that in [16]. More exactly, the discretization takes uniform triangular meshes on Ω, use continuous piecewise linear velocities on a mesh of width d, and use continuous piecewise linear pressures on a mesh of width $2d$ (this discretization is called as $P_1(h)P_1(2h)$ in [16]). We are then led to the saddle point problem (1.1) with $n = 2\,(\frac{1}{d}-1)^2$ and $m = (\frac{2}{d}-1)^2$. For this problem, f is randomly generated with its elements in $[-1,1]$ and h is set to zero.

In this experiment, we vary τ to see the influence of the inexactitude of the subproblem on the performance of Algorithm 3.1. For each value of τ, we generate 20 random experiments and observe the average performance of Algorithm 3.1. Note that the purpose of this work mainly focuses on the inexact BB method used for computing p_k. For ease in coding, we simply use the (exact) BB method with no preconditioner to solve the subproblem for u_{k+1} (see Step 2 of Algorithm 3.1). It is certain that other methods possibly with some preconditioner can be used to solve the subproblem, for example conjugate gradient methods or other gradient methods. Considering the special structure of A, some other effective iterative or direct methods can be also found. When the (exact) BB method is used for the subproblem, at least one step is computed. The maximal iteration number, INMAX say, for the subproblem is set to 100. If this number is exceeded, the inner solver exits with the point having the minimal residuals. Here we note that some other values were also used for INMAX, for example, 250 and 400. However, the difference between the numerical results is not significant.

The tests on Problem 4.2 and the next problem were done with Matlab 7.1.0. We list the numerical results on Problem 4.2 in Table 4.1, where "# OUT" means the average number of outer iterations, "# In" stands for the average number of inner iterations per outer iteration and "time" is the required average CPU time in second. Listed in the last column is the number of outer iterations required by the inexact Uzawa algorithm with the same value of τ for the problem, which can be found in [16] (only the case $d = 1/32$ is available).

Table 4.1: Results of Algorithm 3.1 for Problem 4.2

d	τ	# OUT	# IN	time	Uzawa
1/16	1/64	89.0	78.9	0.995	-
1/16	1/16	99.4	68.4	0.993	-
1/16	1/4	105.3	56.5	0.911	-
1/16	1	146.4	43.7	1.113	-
1/32	1/64	194.5	99.1	9.832	(500)
1/32	1/16	185.3	98.3	9.245	426
1/32	1/4	185.2	95.4	8.915	427
1/32	1	194.7	86.6	8.839	431

Table 4.1 indicates that as τ increases, the average number of outer iterations increases and the cost for inner iteration decreases. In this experiment, the suggested value for τ is 1/4. From the table, we also see that the inexact BB algorithm requires less than one half of the outer iterations by the inexact Uzawa algorithm. This means that if the same solver is used for the subproblem, the inexact BB algorithm, which does not need to estimate the minimal and maximal eigenvalues of A, is even faster.

Problem 4.3. *This problem comes from a different discretization of the Stokes equations with nonzero diverse of velocity, namely, $h \neq 0$. Specifically, we discrete the velocity with continuous piecewise linear finite elements on uniform triangular mesh of width d as before. However, for the pressure function we use the elements suggested in [7] and [6]. More exactly, a piecewise constant function on the adjacent blocks on uniform square mesh of width d is used:*

$$\phi_{ij} = \begin{cases} 1, & x \in (x_{i-1}, x_i),\ y \in (y_{i-1}, y_i) \\ -1, & x \in (x_i, x_{i+1}),\ y \in (y_{i-1}, y_i) \\ -1, & x \in (x_{i-1}, x_i),\ y \in (y_i, y_{i+1}) \\ 1, & x \in (x_i, x_{i+1}),\ y \in (y_i, y_{i+1})\,. \end{cases}$$

The discretization leads to the saddle point problem (1.1) with $n = 2(\frac{1}{d} - 1)^2$ *and* $m = (\frac{1}{d} - 1)^2$. *To guarantee the positive definiteness of the matrix* $C + BA^{-1}B^T$, *we choose* $C = d\,I_m$ *with* I_m *being the identity matrix in* $\Re^{m\times m}$. *It is easy to see that* C *tends to zero as* d *tends to zero. For this problem, both* f *and* g *are randomly generated with their elments in* $[-1, 1]$.

Table 4.2: Results of Algorithm 3.1 for Problem 4.3

d	τ	# OUT	# IN	time
1/8	1/16	36.4	9.7	0.103
1/8	1/8	34.4	9.6	0.097
1/8	1/4	35.2	9.3	0.102
1/8	1/2	37.8	9.2	0.111
1/16	1/16	41.9	9.8	0.494
1/16	1/8	42.9	9.7	0.503
1/16	1/4	42.2	9.5	0.490
1/16	1/2	44.0	9.4	0.516
1/32	1/16	76.7	9.9	4.953
1/32	1/8	70.6	9.8	4.442
1/32	1/4	72.5	9.7	4.588
1/32	1/2	78.3	9.7	5.075
1/64	1/16	153.8	9.9	57.856
1/64	1/8	144.3	9.9	52.254
1/64	1/4	157.2	9.9	59.388
1/64	1/2	150.9	9.8	55.596

For this problem, we vary the value of τ to be 1/16, 1/8, 1/4 and 1/2. The maximum for the inner iterations, INMAX, is set to 10. The results of Algorithm 3.1 for Problem 4.3 are reported in Table 4.2. From the table, we see that the choice of $\tau = 1/8$ is preferred, but the performance of Algorithm 3.1 is not sensitive to the parameter τ. However, we observed that the performance of Algorithm 3.1 heavily depends on the choice of INMAX. In the case of INMAX= 10, we see from the column # IN of Table 4.2 that the the inner solver often reach INMAX iterations. We found that this is also the case if INMAX= 50. Take $d = 1/32$ and $\tau = 1/8$ as an example. If we set INMAX= 50, the average number of inner iterations is 47.8, whereas the algorithm still needs 68.2 outer iterations on the average. The total time is 8.016, about double of the one by choosing INMAX= 10.

The numerical experiments on Problems 4.2 and 4.3 suggest that the inexact BB algorithm is an efficient alternative to the inexact Uzawa algorithm. On the other hand, due to its nonmonotonic feature, the BB algorithm might not be a good option for the inner solver. Further numerical experiments are still required to understand the behavior of the BB algorithm.

5. Conclusions and discussions

In this paper, we have analyzed the inexact BB method with the inexactitude rules (2.6) and (3.6). The analysis with the rule (2.6) could help us in understanding the (exact) BB method for unconstrained optimization, since the latter can be regarded as an inexact BB method for some quadratic function if x_k tends to x^*. Consequently, we are able to prove that the (exact) BB method is locally R-linearly convergent for twice continuously differentiable functions, a result stronger than the one in [12] for the cyclic BB method. Another interesting point with the BB method is that, in the previous analysis of the BB method, the nonmonoticity is introduced by the choice of stepsize α_k. Since the inexactitude rules (2.6) allows the possibility that $-\tilde{g}_k$ is an uphill search direction, Theorem 3.2 tells us that it would be also fine to introduce some suitable nonmonoticity in choosing search directions without affecting the R-linear convergence.

To adapt the inexact BB algorithm for solving saddle point problems, we also analyzed the rule (3.6) carefully and provide R-linear convergence result in the case. These analyses are based on those with the inexact rule (2.6). However, our theoretical analyses cannot provide a good estimate for either the parameter η in (2.6) or the $\bar{\eta}$ in (3.6), although our numerical experiments show that the latter one can be as large as the one in the inexact Uzawa algorithm. It still remains under study how to estimate the parameters η and $\bar{\eta}$ theoretically. The good solution of this problem is related to the question how to establish theoretical evidence showing that the (exact) BB method is faster than the Uzawa algorithm or the steepest descent method in the any dimension case. Some evidence in low dimensions has been established in [3] and [11].

To solve the saddle point problems (1.1), the references [7], [9] and [2] introduce some preconditioner for the inexact Uzawa method:

$$\begin{aligned} u_{k+1} &= Q^{-1}(f - B^T p) \\ p_{k+1} &= p_k + \bar{Q}^{-1}(Cp_k - Bu_{k+1} + h), \end{aligned} \tag{5.1}$$

where Q and $\bar{Q}$ are some approximation to A and $\bar{A} = BA^{-1}B^T + C$ as before. Some extensions are also made to the case when A, B and C are nonlinear operators. Since the BB method does not need to estimate any eigenvalue of the coefficient matrix and is far better than the Uzawa algorithm or the steepest descent method, it might be worthwhile to study the above issues with the inexact BB methods. As the referees commented, the use of preconditioning is indepensible in building fast and pratical methods (here we should note that the preconditioning technique was used to the BB method first by Molina and Raydan [24]). On the other hand, there have been many contenders of the BB stepsize (1.8), see [19] and [11] for example. Our future work is then to establish an efficient inexact and preconditioning BB-like method for saddle point problems.

Acknowledgements. The authors are very grateful to Professor Marcos Raydan in Universida Central de Venezuela and two anonymous referees for their useful comments and suggestions that improved the quality of this paper greatly.

References

[1] K. Arrow, L. Hurwicz and H. Uzawa, *Studies in Nonlinear Programming*, Stanford University Press, Stanford, CA, 1958.

[2] E. Bansch, P. Morin and R. H. Nochetto, *An Adaptive Uzawa FEM for the Stokes Problem: Convergence without the Inf-Sup Condition*, SIAM J. Numer. Anal., Vol. 40, No. 4, pp. 1207-1229, 2002.

[3] J. Barzilai and J. M. Borwein, *Two-Point Step Size Gradient Methods*, IMA J. Numer. Anal., Vol. 8, pp. 141-148, 1988.

[4] M. Benzi, G. H. Golub and J. Liesen, *Numerical Solution of Saddle Point Problems*, Acta Numerica, Vol. 14, pp. 1-37, 2005.

[5] E. G. Birgin, I. Chambouleyron and J. M. Martínez, *Estimation of the Optical Constants, and the Thickness of Thin Films Using Unconstrained Optimization*, J. Comput. Phys., Vol. 151, pp.862-880, 1999.

[6] J. H. Bramble and J. E. Pasciak, *A Preconditioning Technique for Indefinite Systems Resulting from Mixed Approximations of Elliptic Problems*, Math. of Comp., Vol. 50, pp. 1-17, 1988.

[7] J. H. Bramble, J. E. Pasciak and A. T. Vassilev, *Analysis of the Inexact Uzawa Algorithm for Saddle Point Problems*, SIAM J. Numer. Anal., Vol. 34, No. 3, pp. 1072-1092, 1997.

[8] Z.-H. Cao, *Fast Uzawa Algorithm for Generalized Saddle Point Problems*, Applied Numer. Math., Vol. 46, pp. 157-171, 2003.

[9] X. L. Cheng, *On the Nonlinear Inexact Uzawa Algorithm for Saddle-Point Problems*, SIAM J. Numer. Anal., Vol. 37, No. 6, pp. 1930-1934, 2000.

[10] A. Cauchy, *Méthode générale pour la résolution des systèms d'équations simultanées*, Comp. Rend. Sci. Paris, 25, pp. 536-538, 1847.

[11] Y. H. Dai and R. Fletcher, *On the Asymptotic Behaviour of Some New Gradient Methods*, Mathematical Programming (Series A), Vol. 13, No. 3, pp. 541-559, 2005.

[12] Y. H. Dai, W. Hager, K. Schittkowski and H. Zhang, *The Cyclic Barzilai-Borwein Method for Unconstrained Optimization*, Research report, 2005 (to appear in: IMA J. Numer. Anal.)

[13] Y. H. Dai and L. Z. Liao, *R-Linear Convergence of the Barzilai and Borwein Gradient Method*, IMA J. Numer. Anal., vol. 22, pp. 1-10, 2002.

[14] Y. H. Dai and X. Q. Yang, *A New Gradient Method with An Optimal Stepsize Property*, Computational Optimization and Applications, Vol. 33, pp. 73-88, 2006.

[15] Y. H. Dai, J. Y. Yuan and Y. Yuan, *Modified Two-point Stepsize Gradient Methods for Unconstrained Optimization*, Computational Optimization and Applications, Vol. 22, pp. 103-109, 2002.

[16] H. C. Elman and G. H. Golub, *Inexact and Preconditioned Uzawa Algorithms for Saddle Point Problems*, SIAM J. Numer. Anal., Vol. 31, No. 6, pp. 1645-1661, 1994.

[17] H. C. Elman, D. J. Silvester and A. J. Wathen, Finite Elements and Fast Iterative Solvers: with applications in incompressible fluid dynamics, Oxford University Press, 2005.

[18] R. Fletcher, *On the Barzilai-Borwein Method*, In: Optimization and Control with Applications (L. Qi, K. Teo, X.Q. Yang, eds.), Springer, pp. 235-256, 2005.

[19] A. Friedlander, J. M. Martínez, B. Molina and M. Raydan, *Gradient Method with Retards and Generalizations*, SIAM J. Numer. Anal., Vol. 36, pp. 275-289, 1999.

[20] W. Glunt, T. L. Hayden and M. Raydan, *Molecular Conformations form Distance Matrices*, J. Computational Chemistry, Vol. 14, No. 1, pp. 114-120, 1993.

[21] L. M. Hernández, *Alternating Oblique Projections for Coupled Linear Systems*, Numerical Algorithms, Vol. 38, pp. 285-303, 2005.

[22] Q. Y. Hu and J. Zou, *Two New Variants of Nonlinear Inexact Uzawa Algorithms for Saddle-Point problems*, Numer. Math., Vol. 93, pp. 333-359, 2002.

[23] W. B. Liu and Y. H. Dai, *Minimization Algorithms Based on Supervisor and Searcher Cooperation*, Journal of Optimization Theory and Applications, Vol. 111, No. 2, pp. 359-379, 2001.

[24] B. Molina and M. Raydan, *Preconditioned Barzilai-Borwein Method for the Numerical Solution of Partial Differential Equations*, Numerical Algorithms, Vol. 13, pp. 45-60, 1996.

[25] M. Raydan, *On the Barzilai and Borwein Choice of Steplenth for the Gradient Method*, IMA J. Numer. Anal., Vol. 13, pp. 321-326, 1993.

[26] M. Raydan, *The Barzilai and Borwein Gradient Method for the Large Scale Unconstrained Minimization Problem*, SIAM J. Optim., Vol. 7, pp. 26-33, 1997.

[27] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1962.